

# Request Form for Translation

198  
Translation Branch  
The world of foreign prior art to you.

Translations:

U. S. Serial No. : \_\_\_\_\_

5 Nov 1991

Requester's Name: Lawrence Shrader  
Phone No. : 305-8046  
Fax No. : 746-7239  
Office Location: CPK2 5Y17  
Art Unit/Org. : 2124  
Group Director: Allen McDonald

PTO 2002-5046

S.T.I.C. Translations Branch

Is this for Board of Patent Appeals? No

Date of Request: 9/23/02

Date Needed By: 27 Sept 02

(Please do not write ASAP-indicate a specific date)

Phone: 308-0881  
Fax: 308-0989  
Location: Crystal Plaza 3/4  
Room 2C01

SPE Signature Required for RUSH:

## Document Identification (Select One):

\*\* (Note: Please attach a complete, legible copy of the document to be translated to this form) \*\*

1. Patent

Document No. 03-346706

Language Japanese

Country Code \_\_\_\_\_

Publication Date 5 Nov 91

No. of Pages \_\_\_\_\_ (filled by STIC)

2. Article

Author \_\_\_\_\_

Language \_\_\_\_\_

Country \_\_\_\_\_

3. Other

Type of Document \_\_\_\_\_

Country \_\_\_\_\_

Language \_\_\_\_\_

## Document Delivery (Select Preference):

Delivery to Exmr. Office/Mailbox Date: 10-16-02 (STIC Only)

Call for Pick-up

Date: \_\_\_\_\_ (STIC Only)

To assist us in providing the most cost effective service, please answer these questions:

Will you accept an English Language Equivalent?

Yes (Yes/No)

Will you accept an English abstract?

No (Yes/No)

Would you like a consultation with a translator to review the document prior to having a complete written translation?

No (Yes/No)

Check here if Machine Translation is not acceptable:

(It is the default for Japanese Patents, '93 and onwards with avg. 5 day turnaround after receipt)

## STIC USE ONLY

### Copv/Search

Processor: 16

Date assigned: 9-15

Date filled: 9-15

Equivalent found: \_\_\_\_\_ (Yes/No)

Doc. No.: \_\_\_\_\_

Country: \_\_\_\_\_

Remarks: \_\_\_\_\_

### Translation

Date logged in: \_\_\_\_\_

PTO estimated words: \_\_\_\_\_

Number of pages: 27

In-House Translation Available: \_\_\_\_\_

In-House: \_\_\_\_\_

Translator: \_\_\_\_\_

Assigned: \_\_\_\_\_

Returned: \_\_\_\_\_

Contractor: \_\_\_\_\_

Name: SC

Priority: 5

Sent: 9/26/02

Returned: 10-14-02

PTO 01-[PTO 2002-5046]

Japanese Patent  
Document No. 3-246706

**Programming Device for A Programmable Controller**  
[Programmable Controller Yohno Programming Sohchi]

Tadashi Abe

UNITED STATES PATENT AND TRADEMARK OFFICE  
Washington, D.C. Month Year

Translated by: Schreiber Translations, Inc.

Country : Japan  
Document No. : 3-246706  
Document Type : Patent  
Language : Japanese  
Inventor : Tadashi Abe  
Applicant : FUJI Electric Co., Ltd.  
FUJI FACOM Corp.  
IPC :  
Application Date : February 26, 1990  
Publication Date : November 5, 1991 ✓  
Foreign Language Title : Programmable Controller Yohho  
Programming Sohchi  
English Title : Programming Device for A  
Programmable Controller

Specification

1. Title of the Invention

PROGRAMMING DEVICE FOR A PROGRAMMABLE CONTROLLER

2. What is Claimed is:

A programming device for a programmable controller wherein a source program for the programmable controller is converted to an execution format program, comprising:

source program recording means 1 that records a source program which is inputted by an operator;

table recording means 2 that records first table 2a wherein a global memory label and an address corresponding to said global memory label are defined, and second table 2b wherein a local memory label, an address corresponding to said local memory label and identifies information indicating a program that uses said local memory label are defined;

table generating means 3 that generates said first table 2a and said second table 2b within said second recording means 2; and

address converting means 4 that reads said source program out of said source program recording means 1 and converts

---

<sup>1</sup> Numbers in the margin indicate pagination in the foreign text.

a label used for said source program to a corresponding address while referring to said first table 2a or second table 2b.

### 3. Detailed Description of the Invention

#### [Summary]

The present invention relates to a programming device for a programmable controller, which converts a source program generated by a user (operator) to an execution format program for a programmable controller.

12

The programming device for a programmable controller has a table which converts not only a global memory label but also a local memory label to an address so that the address of the local memory can be described with a label at the time of coding the source program.

The programming device for a programmable controller, wherein the source program for a programmable controller is converted to an execution format program, comprises a source program recording means that records a source program which is inputted by an operator, a table recording means, which records first table wherein a global memory label and an address corresponding to its global memory label are defined and a second table wherein a local memory label, an address corresponding to its local memory label and identifies information indicating a

program that uses its local memory label are defined, a table generating means, which generates the aforementioned first table and the aforementioned second table within the aforementioned second recording means, and an address converting means, which reads the aforementioned source program out of the aforementioned source program recording means and converts a label used for the aforementioned source program to a corresponding address while referring to the aforementioned first table or the aforementioned second table.

#### [Field of the Invention]

The present invention relates to a programmable controller, and particularly, to a programming device for a programmable controller converting a source program generated by a user (operator) to an execution format program for a programmable controller.

#### [Description of the Prior Arts]

A programmable controller (hereinafter PC), which contains a microcomputer and is a sequencer capable of easily changing the control procedures, is actively used not only for the control of automated production lines but also as a core part determining the working procedure of various automated devices.

In this type of source program for a PC, an address of

program that uses its local memory label are defined, a table generating means, which generates the aforementioned first table and the aforementioned second table within the aforementioned second recording means, and an address converting means, which reads the aforementioned source program out of the aforementioned source program recording means and converts a label used for the aforementioned source program to a corresponding address while referring to the aforementioned first table or the aforementioned second table.

#### [Field of the Invention]

The present invention relates to a programmable controller, and particularly, to a programming device for a programmable controller converting a source program generated by a user (operator) to an execution format program for a programmable controller.

#### [Description of the Prior Arts]

A programmable controller (hereinafter PC), which contains a microcomputer and is a sequencer capable of easily changing the control procedures, is actively used not only for the control of automated production lines but also as a core part determining the working procedure of various automated devices.

In this type of source program for a PC, an address of

the memory of the PC is usually described by a label at the time of coding so that the content of the source program can be easy to understand and the stored address can be freely set after the source program is generated, converted to an execution format program for a PC and stored in the memory. When the source program is converted to an execution format program for a PC, the label in the source program is converted to an address referring to the table corresponding label ? address shown in Figure 4. Adversely, when the execution format program stored in the memory of a PC is displayed in the CRT display and the like in a source program format, the address is converted to a corresponding label referring to the table corresponding to label ? address. That is, in the example shown in Figure 4, labels "ABCD0", "ABCD1" and "ABCD2" are converted to "DI0000", "DI0001" and "DI0002" respectively.

Note that the aforementioned table corresponding to label ? address shown in Figure 4 is usually generated by an operator on the screen.

[Problems to Be Solved by the Present Invention]

The aforementioned table corresponding to label ? address shown in Figure 4 is a table defining the correspondence between an address of the global memory and a label of the global memory, and not a table defining that of an address of the local memory and a label of the local memory.



Among PC memories, there are: the global memory to which all the programs can have an access and the local memory to which only individual programs have an access including the work areas dynamically reserved for each program at the time of execution of the programs and the parameters (arguments) of the function module when they are called by each program and passed over from the upper layer programs (parent programs).

/3

The conventional programming devices cannot define a label in the address of the local memory because of the reasons described below.

As shown in Figures 5a and 5b, the addresses of the local memories are expressed by address parts 22 and 32, which show the addresses (relative positions) within the local memories of identification parts 21 and 31, which indicate the local memories.

Figure 5a indicates a description of an address within the work area and the first two letters "TM" are identification part 21 representing a label of the work area and the consecutive four digit figure "0010" is an address (relative address) 22 within the work area when the work area is placed on the memory. Figure 5b indicates a description of a parameter of the function module when the function module is passed over from the upper layer program and the first two letters "PM" are identification

part 31 and the consecutive four digit figure "0002" is parameter number 32.

In a program, when an address accesses to the work area of "TM", its address description becomes "TM0010" no matter what programs are used, but a physical address when the aforementioned work area is placed on the memory differs from one program to another. This is because when a program is executed, the work area is dynamically reserved within a vacant area of the memory at the time. This also applies to the parameter of the function module. That is, in order to describe as a label the address of the local memory, such as the address of the work area or the parameter of the function module, at the time of coding of a source program, it is necessary to allocate different physical addresses when the address of the local memory is allocated on the memory, even though the name of the label describing the address of the aforementioned local memory is used for a plurality of programs.

The purpose of the present invention is to provide a table for converting a label not only to an address of the global memory but also to that of the local memory so that at the coding of a source program the address of the local memory can be also described as a label.

[Means to Solve the Problems]

Figure 1 illustrates an explanatory view of the topics of the present invention.

In Figure 1, 1 represents a source program recording means, which records a source program (outer form program) inputted by an operator through means such as a keyboard. This source program recording means comprises semiconductor memory such as RAM (Random Access Memory) or magnetic memory such as a floppy disc or a hard disc.

2 represents a table recording means, which records first table 2a and second table 2b that are generated by after-mentioned table generating means 3. The aforementioned first table 2a is, for example, a table format defining a global memory label, wherein an address of the global memory, to which all the programs can have an access, is expressed in a label format within the aforementioned source program, and an address corresponding to the global memory label. Second table 2b is a table defining a local memory label, wherein an address of the local memory such as the work area dynamically reserved at the time of executing programs, to which only individual programs can be accessed, or the parameters of the function module when the function module is passed over from the upper layer program (parent program), is expressed in a label format within the aforementioned source program, and defining identifying information indicating the programs that use the local memory

label. Table recording means 2 recording first table 2a and second table 2b described above comprises semiconductor memory such as RAM or magnetic memory such as a floppy disc or a hard disc.

3 represents a table generating means, which generates the aforementioned first table 1 and second table within the aforementioned table recording means.

/4

An operator executes an input for generating the aforementioned first table 2a and second table 2b by command input method using means such as a keyboard, or by screen input method looking at the aforementioned first table 2a and second table 2b displayed on display devices such as CRT displays.

4 represents an address converting means, which reads statements of a source program out of source program recording means 1, sequentially one by one from the beginning of the program, and, when a label is contained in a readout statement, searches first table 2a and second table 2b and converts the label (specified label) to a corresponding address. This table search is conducted, for example, in the order of second table and first table, and in a second table search, a search is conducted with the aforementioned specified label and a program containing the specified label as the key to find an address matching the aforementioned specified label with the

aforementioned program. If a matching address is found, the aforementioned specified label is identified as a local memory label and the found address becomes a converted address.

If second table 2b does not have any of the corresponding labels, first table 2a is searched with the aforementioned specified label as the key, and a label matching the aforementioned specified label is searched. If a label matching the aforementioned specified label is found, the aforementioned specified label is identified as a global memory label and an address corresponding to the found label becomes a converted address.

#### [Function]

In the present invention, when an operator executes an input for instructing a global memory label and an address corresponding to the global memory label, first table 2a for the aforementioned global memory label is automatically generated by table generating means 3. When the operator executes an input for instructing a local memory label, identifying information of a program using the local memory label and an address corresponding to the local memory label, second table 2b for the aforementioned local memory label is automatically generated by table generating means 3.

In this way, an operator can generate not only first

table 2a for converting a global memory label to an address corresponding to a program which will be generated, but also second table 2b for converting a local memory label, which corresponds to a program used by the operator, to an address. Therefore, when generating a source program (outer form program), the operator can describe an address of the local memory in a label.

That is, when a source program memorized in source program recording means 1 is converted to an execution format program, a label used in the source program is converted to an address for the execution format program by address converting means 4, and address converting means 4 converts a local memory label to an address for the execution format program referring to second table 2b. In second table 2b, the local memory label is memorized corresponding to identifying information of the program using the label and the address corresponding to the label. Therefore, in a plurality of programs, even if a local memory label having the same name as others is used (described), by referring to second table 2b, address converting means 4 can precisely convert the aforementioned local memory label having the same name to an address of the local memory corresponding to each program in accordance with an operator's instruction. That is, when an operator generates a source program, he can describe an address of the local memory with a label.

[Embodiments]

The embodiments of the present invention will be described below by referring to the figures.

[Architecture]

#### System Architecture

Figure 2 illustrates a block diagram showing a system architecture of an embodiment of the programming device of the present invention.

/5

In Figure 2, input part 41 includes a keyboard executing input of a source program (outer form program) or information corresponding to label ? address (information defining a label and an address corresponding to the label) and a microprocessor conducting analysis of the data inputted by the keyboard. Input part 41 writes the inputted source program on outer form program memory part 42 consisting of RAM (Random Access Memory) and the like. Display part 43 has a display device consisting of a CRT display, liquid crystal display and the like, which converts the aforementioned source program stored in the aforementioned outer form program memory part to a data in a display format and displays the data in the aforementioned display device.

Input part 41 outputs the information corresponding to label ? address, which is inputted by an operator, to label ?

address defining part 44.

Label ? address defining part 44 analyzes the inputted information corresponding to label ? address and identifies whether the label is the global memory label for the global memory or the local memory label for the local memory. If the label is the global memory label, label ? address defining part 44 writes the label in table corresponding to global memory label ? address 45, and if the label is the local memory label, it writes the label in table corresponding to local memory label ? address 46.

The aforementioned table corresponding to global memory label ? address 45 and table corresponding to local memory label ? address 46 consist of RAM (Random Access Memory) and the like and respectively memorize the information defining the global memory label and the address corresponding to the global memory label and the information defining the local memory label and the address corresponding to the local memory label.

Program converting part 47 includes a microprocessor and, when the aforementioned input part 41 executes input such as the one ordering to convert a prescribed source program stored in the aforementioned outer form program memory part 42 to an execution format program, program converting part 47 reads each statement of the aforementioned source program out of the outer form program memory part 42 sequentially from the beginning of



each statement and, if the statement is an order, program converting part 47 converts it to a machine code, which is the corresponding execution format order, and stores the machine code in execution format program memory part 48, which consists of RAM and the like. If the statement contains a label, program converting part 47 outputs a number (program number) defined for the label and the source program containing the label and being converted to the execution format program, to label ? address converting part 49.

When a program number of the aforementioned label and the source program containing the label is inputted, label ? address converting part 49 refers to table corresponding to local memory label ? address 46 and searches to find whether or not there is an address corresponding to the label. If the address is found, label ? address converting part 49 returns the address corresponding to the label (local memory label) to program converting part 47. If the address is not found, then, label ? address converting part 49 searches to find whether or not there is an address corresponding to the aforementioned label in table corresponding to global memory label ? address 45. If the address is found, label ? address converting part 49 returns the address to program converting part 47. If the address is not found in table corresponding to global memory label ? address 45, label ? address converting part 49 notifies program converting

part 47.

Program converting part 47 writes the address returned from label ? address converting part 49 as the address of execution format program memory part 48.

Note that the aforementioned label ? address defining part 44, program converting part 47 and label ? address converting part 49 can be made from the same microprocessor. Outer form program 42, table corresponding to global memory label ? address 45, table corresponding to local memory label ? address 46 and execution format program memory part 48 can be made from the same semiconductor memory or magnetic memory such as a floppy disc or a hard disc.

/6

#### Architecture of Table Corresponding to Global Memory Label ? Address

Figure 3a is a block diagram of the inside of table corresponding to global memory label ? address 45.

Table 45 has exactly the same structure as that of the conventional table corresponding to label ? address shown in Figure 4 described above, and records several sets of the information consisting of label 45a, a global memory label, and address 45b corresponding to global memory label 45a.

#### Architecture of Table Corresponding to Local Memory Label ? Address 46

Figure 3b is a block diagram of the inside of table corresponding to local memory label ? address 46.

As shown in Figure 3b, table 46 records several sets of the information consisting of label 46b, which is a local memory label, program number 46b, which is the identifying number of a program where local memory label 46b is described and address 46 corresponding to local memory label 46b.

In this way, local memory label 46 records local memory label 46b in correspondence with the program where local memory label 46b is described, therefore, even if a plurality of programs use local memory label 46b that has the same name to describe the address of the local memory, when the programs are loaded with memories, different addresses corresponding to local memory label 46b are assigned to each program.

Therefore, in the coding of a source program, it is possible to describe an address of the local memory with local memory label 46b.

In an embodiment shown in Figure 3b, each source program respectively having "PG0000" and "FM0000" as program numbers 46a defines discretionary addresses of the local memory as local memory label 46b, "ABCD". Here, when the aforementioned source program is converted to an execution format program, physical address 46c of each program, which is "TM0000" (when the program has program number 46a of "PG0000") or "PM0000" (when the program

has program number 46a of "FM0003"), is assigned to the local memory defined by local memory label 46b "ABC0".

[Operation]

Next, the operation of the programming device having the aforementioned structure will be explained.

#### Generation of A Program

A program is generated by an operator inputting a source program from input part 41 by using keyboard and the like.

The source program inputted by an operator (outer form program) is sequentially written by input part 41 on outer form program memory part 42. Addresses of the global memory and local memory are written with labels in this source program.

The source program written on outer form program memory part 42 is read by display part 43 and displayed in real time in display devices such as CRT display and liquid crystal display.

by inputting the command in question from input part 41, it is possible to display the program stored in the aforementioned outer form program memory part 42 in display part 43 as needed.

#### Generation of Table Corresponding to Label ? Address

Table corresponding to global memory ? label 45 and table corresponding to local memory ? label 46, which are shown in Figures 3a and 3b, are generated by an operator from input part 41 inputting a command defining the label indicating the

address of the global memory or that of the local memory.

/7

The command defining the label has a format containing a parameter clearly specifying whether the label is for the global memory or for the local memory (label identifying ID), a label name and an address corresponding to the label name. The command defining local memory label 46b also contains the program number of a program using label 46b.

When the command defining the aforementioned label is inputted, input part 41 identifies whether the label is global memory label 45a or local memory label 46b based on the aforementioned label identifying ID. If the label is global memory label 45a, input part 41 writes one set of the information consisting of label 45a and address 45b, which are assigned by the command, on table converting global memory label ? address 45 shown in Figure 3a in the order of the input. If the label assigned by the command is local memory label 46b, input part 41 writes one set of the information consisting of program number 46a, label 46b and address 46c, which are assigned by the command, on table converting local memory label ? address 46 in the order of the input.

#### Conversion from A Source Program (Outer Form Program) to An Execution Format Program

When converting a source program memorized in outer form

program memory part 42 to an execution format program, an operator inputs a command instructing the conversion of the source program to an execution format program from input part 41.

When the aforementioned command is inputted, input part 41 outputs program number 46a of the program being converted as well as an order to convert the program to an execution format program at program converting part 47.

When the aforementioned order is inputted, program converting part 47 reads a source program containing the assigned program number 46a out of outer form program memory part 42 and converts it to an execution format program.

This program converting procedure is done by reading the source program by the statement sequentially from the beginning, and if the statement has the label, that label and the program number are outputted to label ? address converting part 49.

Label ? address converting part 49 first searches table corresponding to local memory label ? address 46 using the inputted label as the key. If label 46b having the same name as that of the inputted label is found, label ? address converting part 49 determines whether or not program number 46a corresponding to label 46b is the same program number as that of the aforementioned source program. If the number is not the same as the program number in question, label ? address converting part 49 continues searching table corresponding to local memory

label ? address 46 using the inputted label as the key. By conducting these searches, if label ? address converting part 49 finds one set of the data containing program number 46a and label 46b that match the program number and the label received from program converting part 47, label ? address converting part 49 returns the address corresponding to label 46b, that is, the address of local memory label 46b, to program converting part 47.

If any set of the data containing the program number and the label assigned by the aforementioned program converting part 47 is not found in table converting local memory label ? address 46, label ? address converting part 49 searches table corresponding to global memory label ? address 45 using the aforementioned label as the key. If the aforementioned assigned label is found in table 45, label ? address converting part 49 returns address 45b corresponding to the assigned label 45a to program converting part 47. If the aforementioned assigned label is not found in table corresponding to global memory label ? address 45, label ? address converting part 49 notifies program converting part 47.

When program converting part 47 receives the address from label ? address converting part 49, it writes the address in the area of execution format program memory part 48.

When there is an order (instruction) in the read statement, program converting part 47 converts that order to a

corresponding execution format order (machine code) and writes it in the area of the execution format program memory part 48.

/8

The aforementioned procedure is conducted on all the statements of the source program that is memorized in outer form program memory part 42 by program converting part 47, and the aforementioned source program is converted to the execution format program and stored in execution format program memory part 48.

In this way, label 46b indicating the address of the local memory in the source program is converted to an address defined by an operator by program converting part 47, which uses table corresponding to local memory label ? address 46. In this case, local memory label 46b is memorized in table corresponding to local memory label ? address 46 in correspondence with program number 46a, therefore, even if local memory label 46b having the same name is described in a plurality of source programs, the aforementioned label 46b is converted to a correct address 46c.

In the embodiment described above, table corresponding to global memory label ? address 45 and table corresponding to local memory label ? address 46 are generated, and by displaying tables, which are shown in Figures 3a and 3b in the display part, and inputting label 45a, address 45b (in the case of global memory label converting table 45), program number 46a, label 46b



and address 46c (in the case of local memory label converting table 46), the aforementioned converting tables 45 and 46 can be generated.

#### [Effects of the Invention]

As described above, according to the present invention, in a programming device for programmable controller wherein a source program for the programmable controller is converted to an execution format program, an operator can freely generate a first table recording a global memory label and an address corresponding to the label as well as a second table recording a local memory label corresponding to the program using the local memory label and the address corresponding to the local memory label, and when the source program is converted to an execution format program, the local memory label used in the source program is converted to the address of the local memory based on the aforementioned second table.

As a result, an operator can describe (express) the address of the local memory with the label when generating a source program. Further, it is not only possible to generate a source program but also to debug it. Moreover, it is possible to shorten the time for developing the source program. When the aforementioned local memory label is converted to the address, the programming device can determine whether or not the name of

the label matches and whether or not the program, which is being used, matches. Therefore, the local memory label having the same name can be used for a plurality of different programs and the device is advantageous in generating source programs.

#### 4. Brief Description of the Drawings

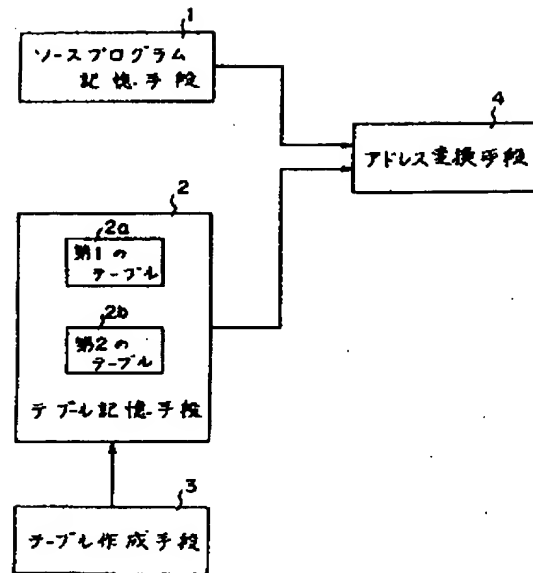
Figure 1 is an explanatory view of the principle of the present invention,

Figure 2 is a system block diagram of an embodiment of the programming device according to the present invention,

Figures 3a and 3b are block diagrams of the insides of table corresponding to global memory label ? address and table corresponding to local memory label ? address,

Figure 4 is a block diagram of the inside of label ? address converting table of the conventional programming device, and

Figures 5a and 5b are diagrams expressing examples of the address of the local memory.



本発明の原理説明図

第 1 図

[Figure 1. Key:]

Explanatory view of the principle of the present invention.

1. source program recording means

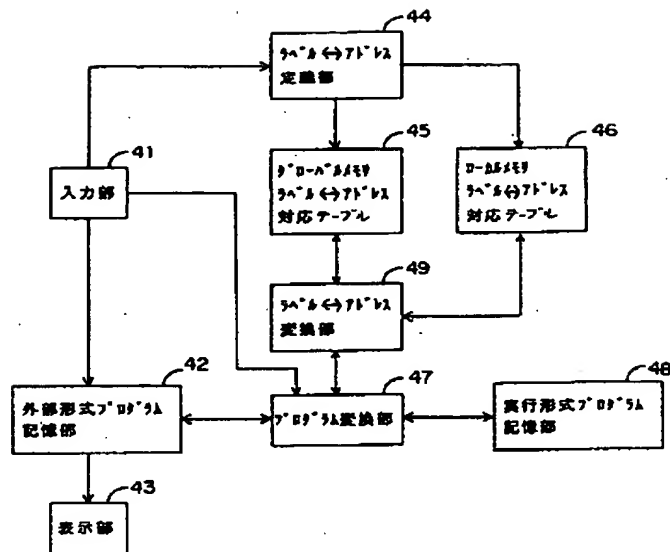
2. table recording means

2a. first table

2b. second table

3. table generating means

4. address converting means



本発明に係る一実施例であるプログラミング装置のシステム構成図

図 2

[Figure 2. Key:]

System block diagram of an embodiment of the programming device according to the present invention.

41. input part

42. outer form program memory part

43. display part

44. label ? address defining part

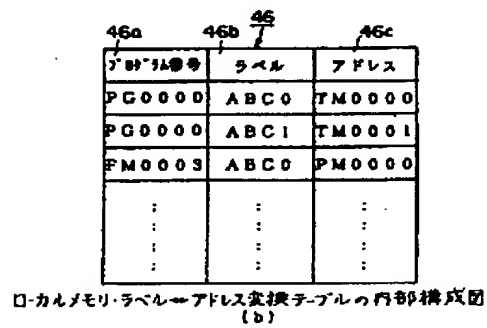
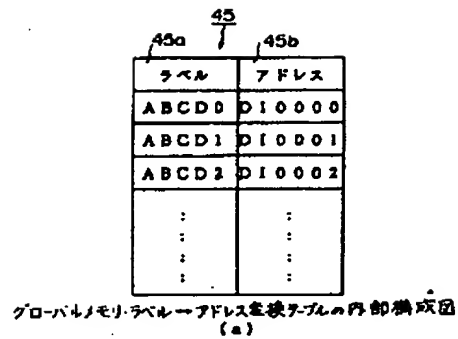
45. table corresponding to global memory label ? address

46. table corresponding to local memory label ? address

47. program converting part

48. execution format program memory part

49. label ? address converting part



[Figure 3. Key:]

a. block diagram of the inside of table corresponding to global memory label ? address

45a. label

45b. address

b. block diagram of the inside of table corresponding to local memory label ? address

46a. program number

46b. label

46c. address

ラベル	アドレス
ABCD0	DI0000
ABCD1	DI0001
ABCD2	DI0002
:	:
:	:
:	:
:	:

従来のプログラミング装置のラベル→アドレス対応  
テーブルの内部構成図

第 4 図

<sup>21</sup>  
<sup>22</sup>  
 TM 0010  
 作業領域のアドレス  
 (a)

<sup>31</sup>  
<sup>32</sup>  
 PM 0002  
 パラメータのアドレス  
 (b)

ローカルメモリのアドレスの表現例  
を示す図

第 5 図

[Figure 4. Key:]

Block diagram of the inside of label ? address converting table  
of the conventional programming device.

[Figure 5a and 5b.]

Diagrams expressing examples of the address of the local memory.

a. address of the work area

b. address of the parameter

CLIPPEDIMAGE= JP403246706A

PAT-NO: JP403246706A

DOCUMENT-IDENTIFIER: JP 03246706 A

TITLE: PROGRAMMING DEVICE FOR PROGRAMMABLE CONTROLLER

PUBN-DATE: November 5, 1991

INVENTOR-INFORMATION:

NAME

ABE, TADASHI

ASSIGNEE-INFORMATION:

NAME

FUJI ELECTRIC CO LTD

FUJI FACOM CORP

COUNTRY

N/A

N/A

APPL-NO: JP02042643

APPL-DATE: February 26, 1990

INT-CL (IPC): G05B019/05

US-CL-CURRENT: 700/83

ABSTRACT:

PURPOSE: To constitute the device so that an address of a local memory can be described by a label at the time of coding of a source program by providing a table for converting the label to the address on not only a global memory but also the local memory.

CONSTITUTION: When an operator executes an input for instructing a global memory label and an address corresponding to its global memory label, a first table 2a for the global memory label is generated automatically by a table generating means 3. Subsequently, when the operator executes an input for instructing a local memory label, discriminating information of a program for using its local memory label, and an address corresponding to its local memory label, a second table 2b for the local memory label is generated automatically by the table generating means 3. In such a way, at the time of generating a source program, the operator can describe (express) the address of the local memory.

COPYRIGHT: (C)1991, JPO&Japio

## ⑫ 公開特許公報(A) 平3-246706

⑤ Int. Cl.<sup>5</sup>

識別記号

庁内整理番号

⑬ 公開 平成3年(1991)11月5日

G 05 B 19/05

A

7740-3H

審査請求 未請求 請求項の数 1 (全10頁)

⑭ 発明の名称 プログラマブル・コントローラ用のプログラミング装置

⑮ 特 願 平2-42643

⑯ 出 願 平2(1990)2月26日

⑰ 発 明 者 阿 部 正 東京都日野市富士町1番地 富士ファコム制御株式会社内  
 ⑱ 出 願 人 富士電機株式会社 神奈川県川崎市川崎区田辺新田1番1号  
 ⑲ 出 願 人 富士ファコム株式会社 東京都日野市富士町1番地  
 ⑳ 代 理 人 弁理士 大 菅 義 之

## 明 細 書

## 1. 発明の名称

プログラマブル・コントローラ用の  
プログラミング装置

## 2. 特許請求の範囲

プログラマブル・コントローラ用のソースプログラムを実行形式のプログラムに変換するプログラマブル・コントローラ用のプログラミング装置において、

オペレータにより入力されたソースプログラムを記憶するソースプログラム記憶手段(1)と、

グローバルメモリ・ラベルとそのグローバルメモリ・ラベルに対応するアドレスが定義されている第1のテーブル(2a)とローカルメモリ・ラベルとそのローカルメモリ・ラベルに対応するアドレス、及びそのローカルメモリ・ラベルを使用するプログラムを示す識別情報とが定義されている第2のテーブル(2b)を記憶するテーブル記

憶手段(2)と、

前記第1のテーブル(2a)及び前記第2のテーブル(2b)を前記第2の記憶手段(2)内に作成するテーブル作成手段(3)と、

前記ソースプログラム記憶手段(1)から前記ソースプログラムを読み出して、前記第1のテーブル(2a)または前記第2のテーブル(2b)を参照しながら、前記ソースプログラムで使用されているラベルを対応するアドレスに変換するアドレス変換手段(4)と、

を具備することを特徴とするプログラマブル・コントローラ用のプログラミング装置。

## 3. 発明の詳細な説明

(概 要)

ユーザ(オペレータ)が作成したソースプログラムをプログラマブル・コントローラ用の実行形式のプログラムに変換するプログラマブル・コントローラ用のプログラミング装置に関し、

グローバルメモリのみならずローカルメモリに



対してもラベルをアドレスに変換するためのテーブルを設け、ソースプログラムのコーディングにおいてローカルメモリのアドレスもラベルで記述できるようにすることを目的とし、

プログラマブル・コントローラ用のソースプログラムを実行形式のプログラムに変換するプログラマブル・コントローラ用のプログラミング装置において、オペレータにより入力されたソースプログラムを記憶するソースプログラム記憶手段と、グローバルメモリ・ラベルとそのグローバルメモリ・ラベルに対応するアドレスが定義されている第1のテーブルとローカルメモリ・ラベルとそのローカルメモリ・ラベルに対応するアドレス、及びそのローカルメモリ・ラベルを使用するプログラムを示す識別情報とが定義されている第2のテーブルを記憶するテーブル記憶手段と、前記第1のテーブル及び前記第2のテーブルを前記第2の記憶手段内に作成するテーブル作成手段と、前記ソースプログラム記憶手段から前記ソースプログラムを読み出して、前記第1のテーブルまたは前

記第2のテーブルを参照しながら、前記ソースプログラムで使用されているラベルを対応するアドレスに変換するアドレス変換手段と、を具備するように構成する。

#### 〔産業上の利用分野〕

本発明はプログラマブル・コントローラ (Programmable Controller) に係り、特にユーザ (オペレータ) が作成したソースプログラムをプログラマブル・コントローラ用の実行形式のプログラムに変換するプログラマブル・コントローラ用のプログラミング装置に関する。

#### 〔従来の技術〕

マイクロコンピュータを内蔵し、制御手順を容易に変更可能なシーケンサであるプログラマブル・コントローラ (以下、P C と略称する) は、自動化生産ラインの制御用のみならず、産業用ロボットなど各種自動化機器の作業手順を決定する心臓部品として現在盛んに使われている。

このようなP C用のソースプログラムは、ソースプログラムの内容を理解し易くするため、さらにはソースプログラム作成後に、そのソースプログラムをP C用の実行形式のプログラムに変換してメモリに格納する際にその格納アドレスを自由に設定できるようにするために、P Cのメモリのアドレスをラベルで表現してコーディングするのが一般的である。ソースプログラム内のラベルは、そのソースプログラムがP C用の実行形式プログラムに変換される際、第4図に示すラベル→アドレス対応テーブルを参照して、アドレスに変換される。また、逆にP Cのメモリに格納されている実行形式のプログラムをソースプログラムの形式でC R Tディスプレイ等に表示する場合には、そのラベル→アドレス対応テーブルを参照して、アドレスを対応するラベルに変換する。すなわち、第4図に示す例では、ラベル「A B C D 0」、「A B C D 1」、「A B C D 2」がそれぞれ「D 1 0000」、「D 1 0001」、「D 1 0002」のアドレスに変換される。

尚、第4図に示すような前記ラベル→アドレス対応テーブルは、一般的には画面上でオペレータにより作成されるようになっている。

#### 〔発明が解決しようとする課題〕

ところで、前記第4図に示すラベル→アドレス対応テーブルは、グローバル・メモリのアドレスとラベルとの対応を定義するテーブルであり、ローカルメモリのアドレスとラベルとの対応を定義するものではなかった。

P Cのメモリには、全てのプログラムがアクセスすることが可能なグローバル・メモリの他に、プログラムの実行時に各プログラム毎に動的に確保される作業領域や、各プログラムからコール (call) されるファンクション・モジュール (Function Module) が上位プログラム (親プログラム) から引き渡されるパラメータ (引数) のような個別のプログラムのみがアクセス可能なローカルメモリがある。

そして、従来のプログラミング装置は、以下に

示す理由によりローカルメモリのアドレスにラベルを定義することができなかった。

ローカルメモリのアドレスは、第5図(a)、(b)に示すようにローカルメモリを示す識別部21、31とローカルメモリ内のアドレス(相対位置)を示すアドレス部22、32により表現される。

同図(a)は、作業領域内のアドレスを表す記述であり、先頭の2文字「TM」は作業領域のラベルであることを表す識別部21であり、それに続く4桁の数字「0010」は作業領域がメモリ上に配置されたときの作業領域内のアドレス(相対アドレス)22である。また、同図(b)は、ファンクション・モジュールが上位プログラムから引き渡されるパラメータを表す記述であり、先頭の2文字「PM」がパラメータのラベルであることを表す識別部31、続く4桁の数字「0002」がパラメータ番号32を表している。

プログラムで、アドレスが「TM0010」の作業領域をアクセスする場合、そのアドレス記述はいずれのプログラムでも「TM0010」となるが、上

記作業領域がメモリ上に配置されたときの物理アドレスは、各プログラムにおいて異なる。何故ならば、作業領域はプログラムの実行の際に、その時点でのメモリの空領域内に動的に確保されるからである。このことは、作業領域内のアドレス「TM0010」は、各プログラムにおいて異なった目的に使用されることを意味している。このことは、ファンクション・モジュールのパラメータについても同様にあてはまる。すなわち、作業領域のアドレスまたはファンクション・モジュールのパラメータ等のようなローカルメモリのアドレスをソースプログラムのコーディング時にラベルで記述できるようにするためには、上記ローカルメモリのアドレスを記述するラベル名が複数のプログラムで使用されても、そのローカルメモリのアドレスをメモリ上にアロケートする(割り当てる)際に、それぞれ異なる物理アドレスに割り当てることができるようにする必要がある。

本発明は、グローバルメモリのみならずローカルメモリに対してもラベルをアドレスに変換する

ためのテーブルを設け、ソースプログラムのコーディングにおいてローカルメモリのアドレスもラベルで記述できるようにすることを目的とする。

(課題を解決するための手段)

第1図は、本発明の原理説明図である。

図中、1はソースプログラム記憶手段であり、キーボード等を介してオペレータが入力したソースプログラム(外部形式プログラム)を記憶する。このソースプログラム記憶手段はRAM(ランダム・アクセス・メモリ)等の半導体メモリ、またはフロッピディスクもしくはハードディスク等の磁気記憶装置で構成される。

2は、テーブル記憶手段であり、後述するテーブル作成手段3によって作成された第1のテーブル2a及び第2のテーブル2bを記憶する。上記第1のテーブル2aは、全てのプログラムからアクセス可能なグローバルメモリのアドレスが前記ソースプログラム内でラベル形式で表現されたグローバルメモリ・ラベルとそのグローバルメモリ

・ラベルに対応するアドレスを定義する、例えば表形式のテーブルである。また、第2のテーブル2bは、個別のプログラムでのみアクセスされるプログラム実行時に動的に確保される作業領域もしくはファンクション・モジュールが上位プログラム(親プログラム)から引き渡されるパラメータ等のようなローカルメモリのアドレスを前記ソースプログラム内でラベル形式で表現するローカルメモリ・ラベルとそのローカルメモリ・ラベルに対応するアドレス、及びそのローカルメモリ・ラベルを使用するプログラムを示す識別情報とが定義されているテーブルである。これら、第1、第2のテーブル2a、2bを記憶するテーブル記憶手段2は、例えばRAM等の半導体メモリもしくはフロッピディスク、ハードディスク等の磁気記憶装置で構成される。

3は、テーブル作成手段であり、オペレータからの指示に基づいて前記第1のテーブル及び前記第2のテーブルを前記テーブル記憶手段2内に作成する。上記第1、第2のテーブル2a、2bを

作成するためのオペレータ入力、例えばキーボード等を介したコマンド入力、又はCRTディスプレイ等の表示装置に表示される上記第1、第2のテーブル2a、2bの表を観ながらの画面入力により行われる。

4は、アドレス変換手段であり、例えばオペレータからの指示に応じて、ソースプログラム記憶手段1からソースプログラムをその先頭から1ステートメントづつ順次読み出し、読み出したステートメントにラベルが含まれている場合には、第1のテーブル2a及び第2のテーブル2bを検索して、そのラベル（指定ラベル）を対応するアドレスに変換する。このテーブル検索は、例えば第2のテーブル、第1のテーブルの順に行われ、第2のテーブル検索においては、上記指定ラベルとその指定ラベルを含むプログラムをキーとして検索が行われ、上記指定ラベル及び上記当該プログラムの2つが一致するアドレスを見つけ出す。そして、一致するアドレスがあれば、上記指定ラベルがローカルメモリ・ラベルであるものと判別し、

その見つけ出したアドレスを変換アドレスとする。

一方、第2テーブル2a内に該当ラベルが無ければ、次に上記指定ラベルをキーとして第1のテーブル2aが検索され、上記指定ラベルと一致するラベルの検索が行われる。そして、上記指定ラベルと一致するラベルが見つかった場合、上記指定ラベルがグローバルメモリ・ラベルであると判別し、その見つけ出したラベルに対応するアドレスを変換アドレスとする。

#### 〔作 用〕

本発明では、オペレータがグローバルメモリ・ラベルとそのグローバルメモリ・ラベルに対応するアドレスを指示する入力を行うと、テーブル作成手段3により上記グローバルメモリ・ラベル用の第1テーブル2aが自動的に作成されるのみならず、ローカルメモリ・ラベル、そのローカルメモリ・ラベルを使用するプログラムの識別情報、及びそのローカルメモリ・ラベルに対応するアドレスを指示する入力を行うと、テーブル作成手段

3により上記ローカルメモリ・ラベル用の第2テーブル2bが自動的に作成される。

このようにして、オペレータは、作成するプログラムに対応して、グローバルメモリ・ラベルをアドレスに変換するための第1のテーブル2aのみならず、使用プログラムと対応づけられたローカルメモリ・ラベルをアドレスに変換するための第2のテーブル2bを作成することができるので、オペレータはソースプログラム（外部形式プログラム）を作成する際、ローカルメモリのアドレスをラベルで記述することが可能となる。

すなわち、ソースプログラム記憶手段1に記憶されているソースプログラムを、実行形式プログラムに変換する際、そのソースプログラム内で使用されているラベルは、アドレス変換手段4により実行形式のプログラム用のアドレスに変換されるが、アドレス変換手段4は、ローカルメモリ・ラベルを第2のテーブル2bを参照して実行形式のプログラム用のアドレスに変換する。この第2のテーブル2bにおいては、ローカルメモリ・ラ

ベルがそのラベルを使用するプログラムの識別情報及びそのラベルに対応するアドレスとに対応づけられて記憶されている。したがって、アドレス変換手段4は、複数のプログラムにおいて、同一名のローカルメモリ・ラベルが使用（記述）された場合でも、第2のテーブル2bを参照することにより、オペレータの指示通りに、上記同一名のローカルメモリ・ラベルを、各プログラムに対応するローカルメモリのアドレスに正しく変換することができる。換言すれば、オペレータはソースプログラム作成の際、ローカルメモリのアドレスをラベルにより記述することが可能となる。

#### 〔実 施 例〕

以下、図面を参照しながら本発明の実施例を説明する。

#### 〔構成〕

##### システム構成

第2図は、本発明に係る一実施例であるプログラミング装置のシステム構成を示すブロック図で

ある。

同図において、入力部41はソースプログラム（外部形式プログラム）やラベル→アドレス対応情報（ラベルとそのラベルに対応するアドレスを定義する情報）の入力を行うキーボード及びそのキーボードから入力されるデータの解析を行うマイクロプロセッサ等から成り、その入力されたソースプログラムをRAM（ランダム・アクセス・メモリ）等から成る外部形式プログラム記憶部42に書き込む。また、表示部43はCRTディスプレイ、液晶ディスプレイ等から成る表示装置を有し、前記外部形式プログラム記憶部に格納されている前記ソースプログラムを表示形式のデータに変換し、前記表示装置に表示する。

また、入力部41はオペレータにより入力されたラベル→アドレス対応情報をラベル→アドレス定義部44に出力する。

ラベル→アドレス定義部44は、その入力されるラベル→アドレス対応情報を解析し、そのラベルがグローバルメモリ用のグローバルメモリ

・ラベルであるかローカルメモリ用のローカルメモリ・ラベルのいずれかであるかを判別し、グローバルメモリ・ラベルであればグローバルメモリ・ラベル→アドレス対応テーブル45に、ローカルメモリ・ラベルであればローカルメモリ・ラベル→アドレス対応テーブル46に書き込む。

上記グローバルメモリ・ラベル→アドレス対応テーブル45、ローカルメモリ・ラベル→アドレス対応テーブル46は、RAM（ランダム・アクセス・メモリ）等から成り、それぞれグローバルメモリ・ラベルとそのグローバルメモリ・ラベルのアドレスを定義する情報、ローカルメモリ・ラベルとそのローカルメモリ・ラベルのアドレスを定義する情報を記憶する。

プログラム変換部47は、マイクロプロセッサ等から成り、前記入力部41から、前記外部形式プログラム記憶部42に格納されている所定のソースプログラムを実行形式のプログラムに変換する旨の命令を、入力した場合には、外部形式プログラム記憶部42から、その当該ソースプログラ

ムの各ステートメント（statement）を先頭から順次読み出し、そのステートメントが命令であれば、対応する実行形式の命令である機械コード（machine code）に変換し、RAM等から成る実行形式プログラム記憶部48に格納する。一方、そのステートメントがラベルを含んでいれば、そのラベル及びそのラベルを含んでいる現在実行形式のプログラムに変換中のソースプログラムに対し定義されている番号（プログラム番号）をラベル→アドレス変換部49に出力する。

ラベル→アドレス対応部49は、上記ラベルとそのラベルを含むソースプログラムのプログラム番号を入力すると、まずローカルメモリ・ラベル→アドレス対応テーブル46を参照し、そのラベルに対応するアドレスが存在するか検索する。そして、存在すればそのラベル（ローカルメモリ・ラベル）に対応するアドレスをプログラム変換部47に返す。また、もし存在しなければ、次に、グローバルメモリ・ラベル→アドレス対応テーブル45内に、上記ラベルに対応するアドレス

（当該アドレス）が存在するか検索する。そして、アドレスが存在すれば、そのアドレスをプログラム変換部47に返す。また、グローバルメモリ・ラベル→アドレス対応テーブル45内にも当該アドレスが存在しなければ、その旨をプログラム変換部47に通知する。

プログラム変換部47は、ラベル→アドレス変換部49から返されたアドレスを、実行形式プログラム記憶部48の当該アドレスに書き込む。

尚、上記ラベル→アドレス定義部44、プログラム変換部47及びラベル→アドレス変換部49は同一のマイクロプロセッサで構成してもよく、また、外部形式プログラム42、グローバルメモリ・ラベル→アドレス対応テーブル45、ローカルメモリラベル→アドレス対応テーブル46、及び実行形式プログラム記憶部48は、同一の半導体メモリもしくはフロッピーディスクやハードディスク等の磁気記憶装置で構成してもよい。

グローバルメモリ・ラベル→アドレス対応テ

テーブル45の構成

第3図(a)は、グローバルメモリ・ラベル→アドレス対応テーブル45の内部構成図である。

このテーブル45は、前述した第4図に示す従来のラベル→アドレス対応テーブルと全く同様な構成となっており、グローバルメモリ・ラベルであるラベル45aとそのグローバルメモリ・ラベル45aに対応するアドレス45bとから成る1組の情報が複数個記憶されている。

ローカルメモリ・ラベル→アドレス対応テーブル46の構成

第3図(b)は、本発明の特徴であるローカルメモリ・ラベル→アドレス対応テーブル46の内部構成図である。

同図(b)に示すように、このテーブル46には、ローカルメモリ・ラベルであるラベル46b、そのローカルメモリ・ラベル46bが記述されているプログラムの識別番号であるプログラム番号46a、及びそのローカルメモリ・ラベル46bに対応するアドレス46cとから成る1組の情報が

複数個記憶されている。

このように、ローカルメモリ・ラベル46内には、ローカルメモリ・ラベル46bが、そのラベル46bが記述されているプログラムと対応づけられて記憶されているので、複数のプログラムがローカルメモリのアドレスの記述に同一名のローカルメモリ・ラベル46bを使用しても、プログラムがメモリにロードされる際には、そのローカルメモリ・ラベル46bに対し、プログラム毎に異なるアドレスが割り当てられる。

したがって、ソースプログラムのコーディングにおいて、ローカルメモリのアドレスをローカルメモリ・ラベル46bで記述することが可能となる。

第3図(b)に示す具体例では、プログラム番号46aがそれぞれ「PG0000」と「FM0003」となっている各ソースプログラムが、共に任意のローカルメモリのアドレスを「ABC0」というローカルメモリ・ラベル46bで定義しているが、上記各ソースプログラムが実行形式のプログラム

に変換される際、「ABC0」というローカルメモリ・ラベル46bで定義されるローカルメモリには、各プログラム毎にそれぞれ「TM0000」(「PG0000」のプログラム番号46aを有するプログラムの場合)、「PM0000」(「FM0003」のプログラム番号46aを有するプログラムの場合)の物理アドレス46cが割り当てられる。

〔動作〕

次に、上記構成のプログラミング装置の動作を説明する。

プログラム作成

プログラム作成は、オペレータが入力部41からキー操作等によりソースプログラムを入力することにより行われる。

オペレータにより入力されたソースプログラム(外部形式プログラム)は、入力部41により順次外部形式プログラム記憶部42に書き込まれる。このソースプログラムには、グローバルメモリ及びローカルメモリのアドレスがラベルで記述されている。

外部形式プログラム記憶部42に書き込まれたソースプログラムは、表示部43により読み出され、CRTディスプレイまたは液晶ディスプレイ等の表示装置にリアルタイムに表示される。

また、入力部41から当該コマンドを入力することにより、上記外部形式プログラム記憶部42に格納されているプログラムを、随時、表示部43の表示装置に表示することができる。

ラベル→アドレス対応テーブルの作成

第3図(a)、(b)に示すグローバルメモリ→ラベル対応テーブル45、ローカルメモリ→ラベル対応テーブル46は、オペレータが入力部41からグローバルメモリまたはローカルメモリのアドレスを示すラベルを定義するコマンドを入力することにより行われる。

このラベルを定義するコマンドは、グローバルメモリなのか、またはローカルメモリのラベルなのかを明示するパラメータ(ラベル識別ID)、並びにラベル名及びそのラベル名に対応するアドレスを含む形式となっている。また、ローカルメ

メモリ・ラベル46bを定義するコマンドは、そのラベル46bを使用するプログラムのプログラム番号も含んでいる。

入力部41は、上記ラベル定義のコマンドを入力すると、前記ラベル識別IDに基づいて、そのラベルがグローバルメモリ・ラベル45aまたはローカルメモリ・ラベル46bのいずれかであるかを判別する。そして、グローバルメモリ・ラベル45aの場合には、コマンドにより指定されたラベル45aとアドレス45bから成る一組の情報を、第3図(a)に示すグローバルメモリ・ラベル→アドレス変換テーブル45に入力順に書き込んでいく。一方、コマンドにより指定されたラベルがローカルメモリ・ラベル46bである場合には、そのコマンドにより指定されたプログラム番号46a、ラベル46b、及びアドレス46cから成る一組の情報を第3図(b)に示すローカルメモリラベル→アドレス変換テーブル46に入力順に書き込んでいく。

ソースプログラム（外部形式プログラム）から

ラベルをキーとして、まずローカルメモリ・ラベル→アドレス対応テーブル46を検索する。そして、入力されたラベルと同一名のラベル46bが存在すれば、そのラベル46bに対応するプログラム番号46aが前記当該ソースプログラムのプログラム番号に等しいか否かを判別する。そして、当該プログラム番号に等しくない場合には、さらに入力されたラベルをキーとして、ローカルメモリ・ラベル→アドレス対応テーブル46の検索を続行する。このような検索処理を行って、プログラム変換部47から受け取ったプログラム番号とラベルに一致するプログラム番号46aとラベル46bを有する一組のデータを見つけた場合には、そのラベル46bに対応するアドレス、すなわちローカルメモリ・ラベル46bのアドレスをプログラム変換部47に返す。

一方、上記プログラム変換部47により指定されたプログラム番号と指定されたラベルを有する一組のデータがローカルメモリ・ラベル→アドレス変換テーブル46内に存在しない場合には、

#### 実行形式のプログラムへの変換

オペレータは、外部形式プログラム記憶部42に記憶されているソースプログラムを実行形式のプログラムに変換する場合には、入力部41から所望のソースプログラムを実行形式のプログラムに変換する旨を指示するコマンドを入力する。

入力部41は、上記コマンドが入力されると、プログラム変換部47に対し、変換するプログラムのプログラム番号46aと共に、実行形式のプログラムに変換する旨の命令を出力する。

プログラム変換部47は、上記命令を入力すると、指定されたプログラム番号46aを有するソースプログラムを外部形式プログラム記憶部42から読み出し、実行形式のプログラムに変換する。

このプログラム変換処理は、当該ソースプログラムをステートメント単位で先頭から順次読み出し、そのステートメントにラベルが含まれていれば、そのラベルと当該プログラム番号をラベル→アドレス変換部49に出力する。

ラベル→アドレス変換部49は、入力された

ラベル→アドレス変換部49は、次に前記指定ラベルをキーとしてグローバルメモリ・ラベル→アドレス対応テーブル45を検索する。そして、そのテーブル45内に前記指定ラベルが存在すれば、その指定ラベル45aに対応するアドレス45bをプログラム変換部47に返す。また、グローバルメモリ・ラベル→アドレス対応テーブル45内にも、前記指定ラベルが存在しない場合には、その旨をプログラム変換部47に通知する。

プログラム変換部47は、ラベル→アドレス変換部49から、アドレスを返された場合には、そのアドレスを実行形式プログラム記憶部48の当該領域に書き込む。

また、プログラム変換部47は、読み出したステートメントの中に命令(Instruction)があった場合には、その命令を対応する実行形式の命令(機械コード)に変換し、実行形式プログラム記憶部48の当該領域に書き込む。

上述した処理が、プログラム変換部47により外部形式プログラム記憶部42に記憶されている

当該ソースプログラムの全てのステートメントに対し行われ、上記当該ソースプログラムが実行形式のプログラムに変換されて、実行形式プログラム記憶部48に格納される。

このように、ソースプログラム内のローカルメモリのアドレスを示すラベル46bが、ローカルメモリ・ラベル→アドレス対応テーブル46を用いて、プログラム変換部47によりオペレータにより定義されたアドレスに変換される。尚、この場合、ローカルメモリ・ラベル→アドレス対応テーブル46内には、ローカルメモリ・ラベル46bがプログラム番号46aに対応されて記憶されているので、同一名のローカルメモリ・ラベル46bが複数のソースプログラム内で記述されていても、正しいアドレス46cに変換される。

尚、上記実施例では、コマンド入力によりグローバルメモリ・ラベル→アドレス対応テーブル45、ローカルメモリ・ラベル→アドレス対応テーブル46を作成するようにしているが、表示部に第3図(a)、(b)に示すような形式の表を表示す

るようにして、表の空欄にラベル45a、アドレス45b(グローバルメモリ・ラベル変換テーブル45の場合)、プログラム番号46a、ラベル46b、及びアドレス46c(ローカルメモリ・ラベル変換テーブル46の場合)を入力して、上記変換テーブル45、46を作成するようにしてもよい。

#### (発明の効果)

以上説明したように、本発明によれば、プログラマブル・コントローラ用のソースプログラムを実行形式のプログラムに変換するプログラマブル・コントローラ用のプログラミング装置において、グローバルメモリ・ラベルをそのラベルに対応するアドレスと共に記憶する第1のテーブル以外にローカルメモリ・ラベルをそのローカルメモリ・ラベルを使用するプログラムとそのローカルメモリ・ラベルに対応するアドレスとに対応づけて記憶する第2のテーブルを、オペレータが自由に作成できるようにし、ソースプログラムを実行形式

のプログラムに変換する際、そのソースプログラムで使用されているローカルメモリ・ラベルを上記第2のテーブルに基づいてローカルメモリのアドレスに変換する。

この結果、オペレータはソースプログラムを作成する際、ローカルメモリのアドレスをラベルで記述(表現)することが可能となる。さらに、ソースプログラム作成が容易になると共に、そのデバッグ作業も容易となる。さらに、ソースプログラムの開発時間の短縮化も可能となる。また、上記ローカルメモリラベルをアドレスに変換する際には、ラベル名のみならず使用プログラムが一致するかも判別して行われるので、同一名のローカルメモリ・ラベルを複数の異なるプログラムで使用することができ、ソースプログラム作成上、便利である。

#### 4. 図面の簡単な説明

第1図は本発明の原理説明図、

第2図は本発明に係る一実施例であるプログラ

ミング装置のシステム構成図、

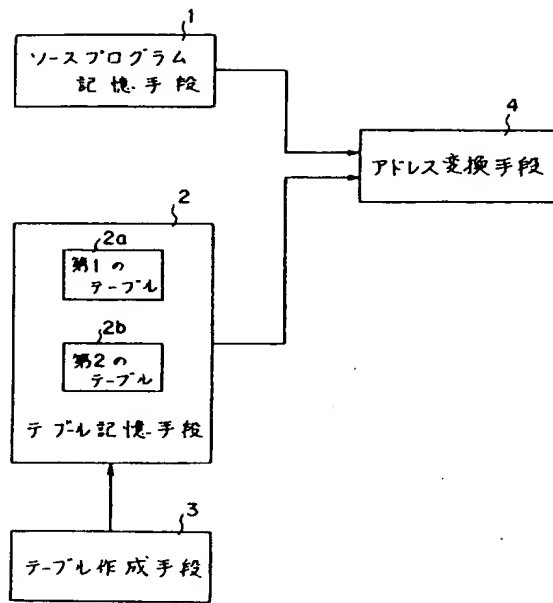
第3図(a)、(b)はそれぞれグローバルメモリ・ラベル→アドレス対応テーブル、ローカルメモリ・ラベル→アドレス対応テーブルの内部構成図、

第4図は従来のプログラミング装置のラベル→アドレス変換テーブルの内部構成図、

第5図(a)、(b)はローカルメモリのアドレスの表現例を示す図である。

- 1・・・ソースプログラム記憶手段、
- 2・・・テーブル記憶手段、
- 2a・・・第1のテーブル、
- 2b・・・第2のテーブル、
- 3・・・テーブル作成手段、
- 4・・・アドレス変換手段。

特許出願人 富士電機株式会社  
同 上 富士ファコム制御株式会社



本発明の原理説明図

第 1 図

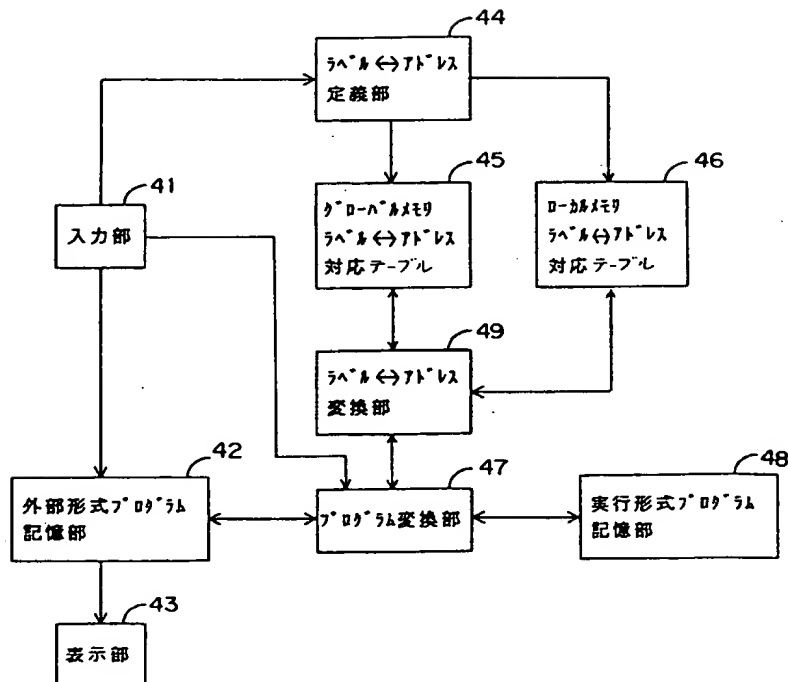
45	
45a	45b
ラベル	アドレス
ABCD0	DI0000
ABCD1	DI0001
ABCD2	DI0002
:	:
:	:
:	:
:	:

グローバルメモリラベル→アドレス変換テーブルの内部構成図  
(a)

46a	46b	46c
プログラム番号	ラベル	アドレス
PG0000	ABC0	TM0000
PG0000	ABC1	TM0001
FM0003	ABC0	PM0000
:	:	:
:	:	:
:	:	:
:	:	:

ローカルメモリラベル→アドレス変換テーブルの内部構成図  
(b)

第 3 図



本発明に係る一実施例であるプログラミング装置のシステム構成図

第 2 図



ラベル	アドレス
ABCD0	DI0000
ABCD1	DI0001
ABCD2	DI0002
:	:
:	:
:	:
:	:

従来のプログラミング装置のラベル⇔アドレス対応  
テーブルの内部構成図

第 4 図

21 22  
TM 0010  
作業領域のアドレス  
(a)

31 32  
PM 0002  
パラメータのアドレス  
(b)

ローカルメモリのアドレスの表現例  
を示す図

第 5 図